12th International Conference on Information Fusion
Seattle, WA, USA, July 6-9, 2009

Robust Kernel-Based Object Tracking with Multiple Kernel Centers*

Shuo Zhang
ECE Department
University of Connecticut
Storrs, CT, U.S.A.
shuo.zhang@engr.uconn.edu

Yaakov Bar-Shalom
ECE Department
University of Connecticut
Storrs, CT, U.S.A.
ybs@engr.uconn.edu

Abstract – *Visual tracking in the real world is challenging with unavoidable background interference, target orientation variations and scale changes. Spatial information needs to be exploited to increase robustness; however, current methods such as "Spatiogram" suffer from the large complexity of spatial covariance calculation. Recently, joint distribution representation has been used to estimate target orientation and scale, but this representation is at the expense of losing position localization information. A new framework is proposed for target model representation by employing multiple kernel centers (MKC) within the kernel window. By employing MKC, spatial information is implicitly embedded. Steepest gradient ascent is used to track the target position, orientation and scale simultaneously. Using an adaptive stepsize in the gradient ascent iteration, the proposed method inherits the desirable properties of the mean shift approach and shows a fast convergence rate. The experimental results in several challenging scenarios demonstrate its robustness and superiority to previous technique.*

Keywords: Visual tracking, kernel, mean shift.

1 Introduction

Object tracking based on visual features such as color and texture have great flexibility to track rigid and non-rigid objects. Extensive work has been done in this area [1, 6, 5, 9], but it is still challenging in the presence of background interference, orientation and scale changes, which usually lead to losing the targets. For a recent survey of object track methods, see [12].

The background-weighted histogram is employed to select the salient parts in target representation [5]. This method requires precalculating the background feature representation around a region which is usually much larger than the target area. Higher-order moments in target representation are used to increase the robustness in tracking [3]. Each bin in the feature space is spatially weighted by the mean and covariance of the locations of the pixels that contribute to that bin, however, the calculation of the mean and covariance is a burden to the complexity.

Multiple kernels are used by introducing the roof kernel [7, 8] based on the SSD (sum of squared differences) measure. The drawback of this representation is that it tends to bring extra noise along the "roof" direction. Also, this approach is not as efficient as the mean shift method due to the complexity of the Newton-style iterations it requires.

Recently, the joint distribution representation has been used by employing the mean shift procedure to estimate target position, orientation and scale simultaneously [11]. One drawback of this approach is that the capability of estimating target orientation is at the expense of losing localization information in the target representation. When the joint distribution is adopted, the kernel function assigns smaller weight to the pixels farther from the orientation direction, where the pixels are valuable for target representation. Another problem is treating scale as a variable. The normalization factor in the target model is independent of the kernel center and orientation, so the mean shift method can be carried out, however, the normalization factor depends on the target scale and it is no longer a constant when scale is treated as a variable, so employing the mean shift method does not guarantee convergence any more.

In this paper we propose a new framework for target model representation. Multiple kernel centers (MKC) are employed inside the kernel window to form an augmented target model. The resulting MKC model contains both the orientation and scale information, which is not possessed by the single kernel center (SKC) model. Also, spatial constrains are implicitly embedded. The orientation and scale estimates are given using steepest gradient ascent. By employing an adaptive stepsize, the proposed method inherits the desirable property of the mean shift algorithm and shows a fast convergence rate. The main contribution is that the paper gives a new approach for building target appearance model and provides target location, orientation and scale estimates simultaneously by using steepest gradient ascent with an adaptive stepsize. Comparisons with [11], which is

*Research sponsored under ONR N00014-07-1-0131 and ARO W911NF-06-1-0467 grants. Proc. 12th International Conference on Information Fusion, Seattle, WA, July 2009.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUL 2009		**06-07-2009 to 09-07-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Robust Kernel-Based Object Tracking with Multiple Kernel Centers	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **ECE Department,University of Connecticut, , Storrs,CT**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002299. Presented at the International Conference on Information Fusion (12th) (Fusion 2009). Held in Seattle, Washington, on 6-9 July 2009. U.S. Government or Federal Rights License.

14. ABSTRACT
Visual tracking in the real world is challenging with unavoidable background interference, target orientation variations and scale changes. Spatial information needs to be exploited to increase robustness; however, current methods such as ?Spatiogram? suffer from the large complexity of spatial covariance calculation. Recently, joint distribution representation has been used to estimate target orientation and scale, but this representation is at the expense of losing position localization information. A new framework is proposed for target model representation by employing multiple kernel centers (MKC) within the kernel window. By employing MKC, spatial information is implicitly embedded. Steepest gradient ascent is used to track the target position, orientation and scale simultaneously. Using an adaptive stepsize in the gradient ascent iteration, the proposed method inherits the desirable properties of the mean shift approach and shows a fast convergence rate. The experimental results in several challenging scenarios demonstrate its robustness and superiority to previous technique.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Public Release**	**8**	

the most recent algorithm in the literature that can handle target location, orientation and scale, show the superiority of our new approach.

The paper is organized as follows. Section 2 presents the MKC model. Section 3 presents the MKC algorithm with location estimation only. Section 4 describes the MKC algorithm incorporated with orientation and scale estimation. Section 5 shows the experimental results. Conclusions are given in Section 6.

2 Target model

We shall introduce the MKC model and describe the normalization issues which are important in the MKC scenario.

2.1 MKC model

Given a kernel described by a convex and a monotonic decreasing kernel profile $k(x)$, the traditional target model q_u is given by

$$q_u = C \sum_{i=1}^{n} k(\|\mathbf{x}_i\|^2) \delta_{\chi(\mathbf{x}_i),u} \tag{1}$$

where the summation is over the pixels in the target region (assumed to have been segmented by an operator in the initial frame), δ is the Kronecker delta function, $\chi : \mathcal{R}^2 \rightarrow \{1, \ldots, m\}$ maps the pixel at location $\mathbf{x}_i$ to the quantized feature, u is an element of the finite set of features $\{1, \ldots, m\}$ and C is the normalization constant for satisfying the condition

$$\sum_{u=1}^{m} q_u = 1 \tag{2}$$

and is given by

$$C = \frac{1}{\sum_{i=1}^{n} k(\|\mathbf{x}_i\|^2)} \tag{3}$$

The candidate model with bandwidth h is given by

$$p_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \right) \delta_{\chi(\mathbf{x}_i),u} \tag{4}$$

where $\mathbf{y}$ is both the centroid of the target region and the kernel center. The normalization constant is

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k(\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\|^2)} \tag{5}$$

This model, computed with a single kernel center (SKC) at the centroid, has limited ability in delineating targets and the resulting mean shift procedure using this SKC model can lead to localization ambiguity [8]. As shown in Fig. 1, the two different targets cannot be discriminated with the SKC target model. Note that the concepts of region centroid and kernel center are different. The region centroid $\mathbf{y}$ represents the location of target and the kernel center indicates where we want to assign large weights to form a target model. For example, we want to put the kernel center on a salient part within the target region to discriminate the background features but the salient part is not necessarily on the region centroid.

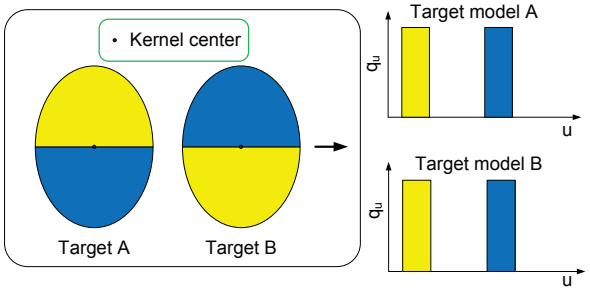

Figure 1: SKC target model.

The idea of MKC is the following: the locations of the region centroid and the kernel center can be different and one can have a number of kernel centers to impose the spatial constraints as long as the target model is normalized. We represent the kernel center $\mathbf{r}_l$ as a function of the region centroid $\mathbf{y}$, rotated angle ϕ (counterclockwise) and bandwidth h by

$$\mathbf{r}_l(\mathbf{z}) \quad = \quad \mathbf{y} + h\Delta\mathbf{r}_l(\phi) \tag{6}$$

where $\mathbf{z} = [y \;\; \phi \;\; h]^T$ and

$$\Delta\mathbf{r}_l(\phi) = d_l \begin{bmatrix} \cos(\phi + \psi_l) \\ \sin(\phi + \psi_l) \end{bmatrix} \tag{7}$$

l represents the lth kernel center and constants d_l, ψ_l are its initial distance and angle in polar coordinates with respect to the centroid $\mathbf{y}$. In view of this, the MKC model can be expressed as

$$q_u = C \sum_{i=1}^{N} \sum_{l=1}^{L} k(\|\mathbf{r}_l - \mathbf{x}_i\|^2) \delta_{\chi_l(\mathbf{x}_i),u} \tag{8}$$

$$p_u(\mathbf{z}) = C(h) \sum_{i=1}^{N} \sum_{l=1}^{L} k\left(\left\| \frac{\mathbf{r}_l(\mathbf{z}) - \mathbf{x}_i}{h} \right\|^2 \right) \delta_{\chi_l(\mathbf{x}_i),u} \tag{9}$$

where L is the number of kernel centers used, $\chi_l : \mathcal{R}^2 \rightarrow \{[(l-1)m + 1], \ldots, lm\}$ maps $\mathbf{x}_i$ to the quantized feature which is calculated from the lth kernel center and u is an element in the finite set $\{1, \ldots, Lm\}$. For the convenience of later derivations, we substitute N for n_h, where N represents the number of all the pixels in a given frame. This is equivalent to the original form because the pixels outside the kernel window do not contribute to the model and N is independent of h in this notation. Since the bandwidth h is treated as a variable, the normalization factor is a function of h, denoted as $C(h)$. Note that $C(h)$ is independent of $\mathbf{y}$ and ϕ given the kernel centers. Imposed by the condition $\sum_{u=1}^{Lm} q_u = 1$ and $\sum_{u=1}^{Lm} p_u(\mathbf{z}) = 1$, C and $C(h)$ are given by

$$C = \frac{1}{\sum_{i=1}^{N} \sum_{l=1}^{L} k(\|\mathbf{r}_l - \mathbf{x}_i\|^2)} \tag{10}$$

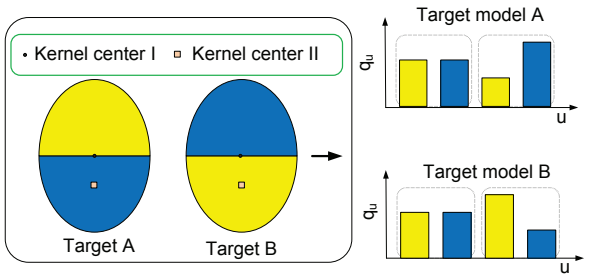

Figure 2: MKC model with two kernel centers.

$$C(h) = \frac{1}{\sum_{i=1}^{N}\sum_{l=1}^{L} k\left(\left\|\frac{\mathbf{r}_l(\mathbf{z})-\mathbf{x}_i}{h}\right\|^2\right)} \tag{11}$$

Note that by employing MKC, the number of quantized features m remains the same but the finite set used to delineate the target model is augmented from $\{1,\ldots,m\}$ to $\{1,\ldots,Lm\}$, where the spatial information is now embedded via the MKC. We use the same example but add another kernel center as shown in Fig. 2. The two targets are discriminated by the MKC model which embodies spatial constraints.

2.2 Scaled radius

An ellipse is employed here to represent the target region. To accommodate the kernel profile representation, the ellipse region should be normalized like a unit circle [5]. The normalized distance $\sigma_{i,l}$ from the pixel $\mathbf{x}_i = [x_i \;\; y_i]^T$ to the lth kernel center $\mathbf{r}_l = [r_x \;\; r_y]^T$ can be represented as,

$$\sigma_{i,l} = \frac{\|\mathbf{x}_i - \mathbf{r}_l\|}{R(\theta,h)} \tag{12}$$

where $\theta = \arctan\frac{y_i - r_y}{x_i - r_x}$ and $R(\theta,h)$ is the scaled radius from the kernel center to the pixel on the ellipse contour which passes through $\mathbf{x}_i$ with angle θ from the horizontal. It can be shown $R(\theta,h)$ is proportional to h given θ due to the geometry similarity. Therefore, $R(\theta,h)$ can be rewritten as

$$R(\theta,h) = R_0(\theta)h \tag{13}$$

where $R_0(\theta)$ is the scaled radius at $h = 1$. The goal is to find $R_0(\theta)$ given $\mathbf{x}_i$ and $\mathbf{r}_l$.

To calculate $R_0(\theta)$ of a current ellipse centered at $\mathbf{y} = [o_x \;\; o_y]^T$, we rotate the ellipse back to its initial position (Fig. 3). Without loss of generality, we assume the initial ellipse ($h = 1$) is given by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{14}$$

where a, b are the semi-axes along the x-axis and y-axis, respectively. The relative position $\Delta\mathbf{x}_i' = [\Delta x_i' \;\; \Delta y_i']^T$ of the pixel $\mathbf{x}_i$ with respect to $\mathbf{y}$ after rotation is given by,

$$\begin{bmatrix} \Delta x_i' \\ \Delta y_i' \end{bmatrix} = \begin{bmatrix} \cos\phi' & -\sin\phi' \\ \sin\phi' & \cos\phi' \end{bmatrix} \begin{bmatrix} x_i - o_x \\ y_i - o_y \end{bmatrix} \tag{15}$$

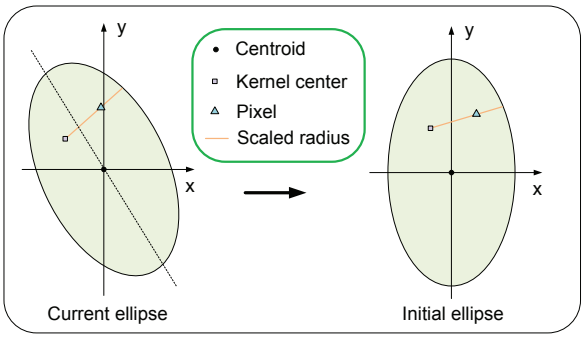

Figure 3: Ellipse Rotation

where $\phi' = -\phi$ is the rotation angle (counterclockwise). Since following rotation, the distance between any two points remains unchanged, we have

$$R_0(\theta) = R_0(\theta') \tag{16}$$

where $\theta' = \arctan\frac{\Delta y_i' - h\Delta r_y'}{\Delta x_i' - h\Delta r_x'}$ and $\Delta\mathbf{r}_l' = [\Delta r_x' \;\; \Delta r_y']^T$ is the relative position of the kernel center with respect to $\mathbf{y}$ in the initial ellipse. Rewrite the initial ellipse in the polar coordinates as

$$\begin{bmatrix} x \\ y \end{bmatrix} = R_0(\theta') \begin{bmatrix} \cos\theta' \\ \sin\theta' \end{bmatrix} + \begin{bmatrix} \Delta r_x' \\ \Delta r_y' \end{bmatrix} \tag{17}$$

From (14) and (17), we obtain

$$\begin{aligned} R_0(\theta') \;=\; & [-b^2\Delta r_x'\cos\theta' - a^2\Delta r_y'\sin\theta' \\ & +(2a^2b^2\Delta r_x'\Delta r_y'\sin\theta'\cos\theta' + a^4b^2\sin^2\theta' + a^2b^4 \\ & \cdot\cos^2\theta' - a^2b^2\cos^2\theta'\Delta r'^2_y - a^2b^2\sin^2\theta'\Delta r'^2_x)^{\frac{1}{2}}] \\ & /(b^2\cos^2\theta' + a^2\sin^2\theta') \end{aligned} \tag{18}$$

Therefore, the normalized distance can be given in an equivalent form by

$$\sigma_{i,l} = \frac{\|\Delta\mathbf{x}_i' - h\Delta\mathbf{r}_l'\|}{R_0(\theta')h} \tag{19}$$

Note that, given the kernel centers in the initial ellipse, $\Delta\mathbf{r}_l'$ is always fixed. In particular, for $\Delta\mathbf{r}_l' = [0 \;\; 0]^T$, (19) reduces to

$$\sigma_{i,l} = \sqrt{\left(\frac{\Delta x_i'}{ah}\right)^2 + \left(\frac{\Delta y_i'}{bh}\right)^2} \tag{20}$$

3 MKC algorithm with location estimation only

We employ the MKC model to form the similarity function defined by the Bhattacharyya Coefficient [5] as

$$\rho(\mathbf{z}) = \sum_{u=1}^{Lm}\sqrt{p_u(\mathbf{z})q_u} \tag{21}$$

To illustrate the relationship between the SKC model and the MKC model, we treat ϕ and h as constants at first. To find the mode of the similarity function, several optimization techniques can be used. However, the major concern is the complexity and the convergence rate. Since evaluating the Hessian matrix is computationally expensive, we employ the steepest gradient ascent to construct the algorithm. The crucial issue here is how to find a suitable stepsize, since a stepsize that is too large will lead to divergence and a stepsize that is too small will result in slow convergence. The mean shift procedure is actually a gradient ascent method with an adaptive stepsize [4]. Therefore, we shall investigate the mean shift stepsize first.

3.1 Mean shift stepsize

Using similar notation as in [4], the kernel density estimate (KDE) is given by

$$f_{h,K}(\mathbf{x}) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \tag{22}$$

where $c_{k,d}$ is the normalization constant, d is the dimension of $\mathbf{x}$, and $k(x)$ is the profile of kernel $K(x)$ with the relationship

$$K(\mathbf{x}) = c_{k,d} k(\|\mathbf{x}\|^2) \tag{23}$$

Define the derivative

$$g(x) = -k'(x) \tag{24}$$

Then the mean shift vector is given by

$$\mathbf{m}_{h,G} = \frac{h^2}{c f_{h,G}(\mathbf{x})} \nabla f_{h,K}(\mathbf{x}) \tag{25}$$

where c is a constant. The function $f_{h,G}(\mathbf{x})$ is the KDE computed with the kernel G by

$$f_{h,G}(\mathbf{x}) = \frac{c_{g,d}}{nh^d} \sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \tag{26}$$

where $G(x)$ is defined as

$$G(\mathbf{x}) = c_{g,d} g(\|\mathbf{x}\|^2) \tag{27}$$

From (25) we can see that the mean shift stepsize $\alpha_{\mathbf{m}}$ is given by

$$\alpha_{\mathbf{m}} = \frac{h^2}{c f_{h,G}(\mathbf{x})} \tag{28}$$

Therefore, in the regions of low-density values, $\alpha_{\mathbf{m}}$ is large while in the regions near the local maxima, $\alpha_{\mathbf{m}}$ is small and the search more refined.

3.2 MKC stepsize

Now consider our problem based on MKC model. Since ϕ and h are treated as constants, (9) reduces to

$$p_u(\mathbf{y}) = C(h_0) \sum_{i=1}^{N} \sum_{l=1}^{L} k\left(\left\| \frac{\mathbf{r}_l(\mathbf{y}) - \mathbf{x}_i}{h_0} \right\|^2 \right) \delta_{\chi_l(\mathbf{x}_i), u} \tag{29}$$

where

$$\mathbf{r}_l(\mathbf{y}) = \mathbf{y} + h_0 \Delta \mathbf{r}_l(\phi_0) \tag{30}$$

and

$$C(h_0) = \frac{1}{\sum_{i=1}^{N} \sum_{l=1}^{L} k\left(\left\| \frac{\mathbf{r}_l(\mathbf{y}) - \mathbf{x}_i}{h_0} \right\|^2 \right)} \tag{31}$$

The linear approximation of $\rho(\mathbf{y})$ defined in (21) is given by [5],

$$\rho(\mathbf{y}) \approx \frac{1}{2} \sum_{u=1}^{Lm} \sqrt{p_u(\mathbf{y}_0) q_u} + \frac{1}{2} \sum_{u=1}^{Lm} p_u(\mathbf{y}) \sqrt{\frac{q_u}{p_u(\mathbf{y}_0)}} \tag{32}$$

where $\mathbf{y}_0$ is the initial centroid in the current frame. Taking the gradient of (32) with respect to $\mathbf{y}$ and using (29), we obtain

$$\nabla \rho(\mathbf{y}) = \frac{C(h_0)}{2} \sum_{i=1}^{N} \sum_{l=1}^{L} \nabla k\left(\left\| \frac{\mathbf{r}_l(\mathbf{y}) - \mathbf{x}_i}{h_0} \right\|^2 \right) w_{i,l} \tag{33}$$

where

$$w_{i,l} = \sum_{u=1}^{Lm} \sqrt{\frac{q_u}{p_u(\mathbf{y}_0)}} \delta_{\chi_l(\mathbf{x}_i), u} \tag{34}$$

The constraint is $p_u(\mathbf{y}_0) > 0$ and the color features should be selected such as to satisfy this constraint. Similarly to (28), the MKC stepsize is given by

$$\alpha = \frac{h_0^2}{C(h_0) \sum_{l=1}^{L} f_l(\mathbf{y})} \tag{35}$$

where

$$f_l(\mathbf{y}) = \sum_{i=1}^{N} w_{i,l} g\left(\left\| \frac{\mathbf{r}_l(\mathbf{y}) - \mathbf{x}_i}{h_0} \right\|^2 \right) \tag{36}$$

Since $f_l(\mathbf{y})$ is the weighted KDE calculated from the lth kernel center, $\sum_{l=1}^{L} f_l(\mathbf{y})$ can be interpreted as a mixture of the estimated probability density characterized by multiple kernel centers. Mixture probability density functions (pdf) are widely used in parametric estimation techniques, such as Gaussian mixture. The sum $\sum_{l=1}^{L} f_l(\mathbf{y})$ can be viewed as the counterpart of the mixture pdf in the nonparametric estimation case (as in the KDE). Note that we omit the normalization constants in $f_l(\mathbf{y})$ and $\sum_{l=1}^{L} f_l(\mathbf{y})$, which are independent of $\mathbf{y}$ given the kernel type. Therefore, the MKC stepsize defined by (35) is adaptive, which makes it possess the same desirable property as the mean shift stepsize.

Employing the steepest gradient ascent by

$$\mathbf{y}^{j+1} = \mathbf{y}^j + \alpha^j \nabla \rho(\mathbf{y}^j) \tag{37}$$

and substituting (35) for α^j, the MKC algorithm[1] is given (after some algebraic manipulations) by

$$\mathbf{y}^{j+1} = \frac{\sum_{i=1}^{N} \sum_{l=1}^{L} [\mathbf{x}_i - h_0 \Delta \mathbf{r}_l(\phi_0)] w_{i,l} g_{i,l}^j}{\sum_{i=1}^{N} \sum_{l=1}^{L} w_{i,l} g_{i,l}^j} \tag{38}$$

[1] At this stage there is no orientation and scale estimation, which will be added in Section 4.

where $g_{i,l}^j$ represents $g(\|\frac{\mathbf{r}_l(\mathbf{y}^j)-\mathbf{x}_i}{h_0}\|^2)$ for short. Note that $\mathbf{y}^j$ cancels out by using this adaptive stepsize.

For convergence analysis, we give the proposition below. The assumptions are that the linear approximation given by (32) is satisfactory and there is at least one nonzero $w_{i,l}$ for each iteration, which are most often valid assumptions between consecutive frames.

Proposition. If the kernel K has a convex and a monotonically decreasing profile, the sequences $\{\mathbf{y}_j\}$ given by (38) converge to $\mathbf{y}^*$, where $\rho(\mathbf{y}^*)$ is the local maximum of the similarity function defined by the Bhattacharyya Coefficient.

The proof is given in the Appendix. Therefore, convergence to $\mathbf{y}^*$ of the MKC Algorithm is guaranteed by using the MKC stepsize given in (35) for fixed orientation and scale.

3.3 Relationship to mean shift procedure

Consider a special case that all the kernel centers are overlapped on the centroid, which means $\Delta\mathbf{r}_l(\phi_0) = 0$. Then, (38) reduces to

$$\mathbf{y}^{j+1} = \frac{\sum_{i=1}^N \mathbf{x}_i w_i g(\|\frac{\mathbf{y}^j-\mathbf{x}_i}{h}\|^2)}{\sum_{i=1}^N w_i g(\|\frac{\mathbf{y}^j-\mathbf{x}_i}{h}\|^2)} \tag{39}$$

where $w_i = Lw_{i,l}$, since $w_{i,l}$ and $g_{i,l}^j$ are independent of l in this case due to the same kernel center. We can see that (39) is exactly the mean shift procedure which uses the SKC model described in [5]. In this case, the MKC model contains the same information as the SKC model, so the two algorithms give the same result.

Since the MKC algorithm given by (38) possesses all the properties of mean shift procedure, such as adaptive stepsize and guaranteed convergence, we can draw the following conclusion: the mean shift procedure is a special case of the MKC algorithm with a single kernel center at the centroid.

4 Incorporation of orientation and scale estimation into the MKC algorithm

For robust tracking, the target region used to delineate the target should be as precise as possible to reject the non-object regions. Therefore, orientation and scale are important parameters to be estimated. Most of the existing approaches restrict themselves to the mean shift framework and suffer from either heuristics or large complexity. Since the MKC model contains both orientation and scale information, we will use it to estimate the orientation and scale.

4.1 Orientation and scale estimation

We employ steepest gradient ascent to optimize target location, orientation and scale simultaneously. Following the procedures discussed in Section 3.2 but without using the linear approximation, the gradient of $\rho(\mathbf{z})$ defined in (21) is given by

$$\nabla\rho(\mathbf{z}) = \sum_{u=1}^{Lm} \frac{\sqrt{q_u}}{2\sqrt{p_u(\mathbf{z})}} \nabla p_u(\mathbf{z}) \tag{40}$$

Since the orientation and scale are not constants any more, stepsize selection is necessary in this case. We employ Armijo rule [2] considering its efficiency and simplicity. The initial stepsize α^0 is a critical parameter for the convergence rate. In view of this, it is natural to use the adaptive stepsize given by (35) to serve as the initial value α^0, which is given by

$$\alpha^0 = \frac{h^2}{C(h)\sum_{l=1}^L f_l(\mathbf{z})} \tag{41}$$

where

$$f_l(\mathbf{z}) = \sum_{i=1}^N w_{i,l} g\left(\left\|\frac{\mathbf{r}_l(\mathbf{z})-\mathbf{x}_i}{h}\right\|^2\right) \tag{42}$$

and

$$w_{i,l} = \sum_{u=1}^{Lm} \sqrt{\frac{q_u}{p_u(\mathbf{z})}} \delta_{\chi_l(\mathbf{x}_i),u} \tag{43}$$

The Armijo Rule stepsize is given by $\alpha^j = \beta^{n'}\alpha^0$, where n' is the first nonnegative integer n that satisfies,

$$\rho(\mathbf{z}^{j+1}) - \rho(\mathbf{z}^j) \geq \lambda\alpha^j \|\nabla\rho(\mathbf{z}^j)\|^2/\gamma^2 \tag{44}$$

where λ, β are fixed scalars satisfying $0 < \lambda < 1, 0 < \beta < 1$. The choice of β is usually from 0.1 to 0.5 [2]. A compensation factor γ is needed here since the distance $\|\frac{\mathbf{r}_l(\mathbf{z})-\mathbf{x}_i}{h}\|$ has been normalized by the scaled radius discussed in Section 2.2. An approximate value is given by $\gamma = \min(a, b)$. Therefore, the increment $\alpha^j\nabla\rho(\mathbf{z}^j)$ of the MKC algorithm can be obtained (after some algebraic manipulations) as

$$\Delta\mathbf{y}^j = \beta^{n'} \frac{\sum_{i=1}^N \sum_{l=1}^L [\mathbf{x}_i - \mathbf{r}_l(\mathbf{z}^j)] w_{i,l}^j g_{i,l}^j}{\sum_{i=1}^N \sum_{l=1}^L w_{i,l}^j g_{i,l}^j} \tag{45}$$

$$\Delta\phi^j = \beta^{n'} \frac{h^j \sum_{i=1}^N \sum_{l=1}^L v_{i,l}^j w_{i,l}^j g_{i,l}^j}{\sum_{i=1}^N \sum_{l=1}^L w_{i,l}^j g_{i,l}^j} \tag{46}$$

$$\Delta h^j = \beta^{n'} \frac{\sum_{i=1}^N \sum_{l=1}^L s_{i,l}^j [w_{i,l}^j - \rho(\mathbf{z}^j)] g_{i,l}^j}{h^j \sum_{i=1}^N \sum_{l=1}^L w_{i,l}^j g_{i,l}^j} \tag{47}$$

where,

$$v_{i,l}^j = (\mathbf{x}_i - \mathbf{y}^j)^T \frac{\partial\Delta\mathbf{r}_l(\phi^j)}{\partial\phi} \tag{48}$$

$$s_{i,l}^j = (\mathbf{x}_i - \mathbf{y}^j)^T (\mathbf{x}_i - \mathbf{r}_l(\mathbf{z}^j)) \tag{49}$$

and $g_{i,l}^j$ represents $g(\|\frac{\mathbf{r}_l(\mathbf{z}^j)-\mathbf{x}_i}{h}\|^2)$ for short. Note that, unlike (34), $w_{i,l}^j$ should be updated as in (43) for each iteration

and $p_u(\mathbf{z}^j)$ cannot be 0 in this case since it is calculated from the updated target region in each iteration. Equations (45)–(47) are carried out iteratively until $\|\mathbf{y}^{j+1} - \mathbf{y}^j\| < \epsilon_{\mathbf{y}}, \|\phi^{j+1} - \phi^j\| < \epsilon_\phi, \|h^{j+1} - h^j\| < \epsilon_h$ are satisfied and $\epsilon_{\mathbf{y}}$ is chosen to satisfy that $\mathbf{y}^{j+1}$ and $\mathbf{y}^j$ are within the same pixel.

Some insight can be obtained for the iterations given above. For $n' = 0$, it can be shown that the centroid iteration given by (45) is the same as (38) except that $w_{i,l}^j$ is updated for each iteration. If all the kernel centers are at the region centroid, which indicates $\Delta\mathbf{r}_l(\phi) = 0$, (46) yields $\phi^{j+1} = \phi^j$. Therefore, the orientation information is not available in this case. Also for $\Delta\mathbf{r}_l(\phi^j) = 0$, (49) reduces to $s_{i,l}^j = \|\mathbf{x}_i - \mathbf{y}^j\|^2$. The norm $\|\mathbf{x}_i - \mathbf{y}^j\|$ is the distance between the pixel $\mathbf{x}_i$ and the centroid. The average distance of all the pixels within the target region represents the target scale. Since $\rho(\mathbf{z}^j)$ is independent of i and l, (47) is actually computing the difference between an unweighted scale and a weighted scale characterized by the weight $w_{i,l}^j$.

Given two consecutive frames, the variations of ϕ and h are always limited and this can be utilized to improve the tracking performance in the high clutter environment. For some threshold $\Delta h_{\max}$ and $\Delta\phi_{\max}$, a feasible solution is given by

$$\phi^{j+1} = \begin{cases} \phi^{j+1} & \text{if } |\phi^{j+1} - \phi_p| \le \Delta\phi_{\max} \\ \phi^j & \text{if } |\phi^{j+1} - \phi_p| > \Delta\phi_{\max} \end{cases} \tag{50}$$

$$h^{j+1} = \begin{cases} h^{j+1} & \text{if } |h^{j+1} - h_p| \le \Delta h_{\max} \\ h^j & \text{if } |h^{j+1} - h_p| > \Delta h_{\max} \end{cases} \tag{51}$$

where ϕ_p, h_p are from the previous frame. A default value for $\Delta h_{\max}$ is $0.1h_p$ and $\Delta\phi_{\max}$ is usually application dependent.

4.2 MKC implementation

In an ideal scenario, without occlusion or background interference, the performance given by the MKC algorithm should be at least no worse than the mean shift method due to the spatial information considered. However, this is not necessarily true in real applications. Some important issues should be taken into consideration before employing the MKC algorithm.

Kernel center selection in MKC algorithm: Intuitively, kernel centers should be far away from each other to provide more discrimination in the target model; however, the noise may increase as the kernel center is away from the centroid, since occlusion or background interference often occurs in the peripheral pixels. Therefore, the kernel centers should be restricted to some region around the centroid. From most situations, the maximum distance to the centroid for the kernel center should be no more than $1/3$ of the minor axis.

Parallel MKC (PMKC) algorithm: Compared to the noise in the SKC model, if the occlusion or background interference is near the "perigee" of the kernel ellipse (with respect to the kernel center), the noise in the MKC model is larger; if it is near the "apogee" of the kernel ellipse, the noise in the MKC model is smaller. In view of this, we propose a procedure called PMKC algorithm:use two sets of MKC on opposing sides within the kernel window and run the two MKC algorithms in parallel. The best result which yields the largest Bhattacharyya coefficient is retained . Though the computational cost is a little higher, it yields a very robust tracking performance.

Figure 4: Comparison of mean shift and MKC algorithm.

5 Experimental results

The RGB color space is quantized into $16\times16\times16$ bins. An Epanechnikov profile

$$k(x) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-x) & \text{if } x \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{52}$$

is employed, where c_d is the unit volume of d-dimensional (2 in our case) sphere. The MKC algorithm is employed by using two kernel centers through all the experiments. One kernel center is on the centroid and the other one is on the axis. Since two different algorithms are described in Section 3 (MKC algorithm with location estimation only) and Section 4 (MKC algorithm with location, orientation and scale estimation) respectively, the experimental results are given in two parts.

5.1 Localization with fixed orientation and scale

The performance of the MKC algorithm given by (38) for fixed orientation and scale, shown in the bottom row of Fig. 4, is compared with the mean shift algorithm (top row of Fig. 4). For $\epsilon_{\mathbf{y}} = 0.7$ (in image coordinates), the average number of iterations is about 3 for both algorithms. We can see the mean shift algorithm yielded ambiguity in the localization due to the background interference while the MKC algorithm, due to its use of two kernel centers, tracked the target correctly.

5.2 Localization with orientation and scale estimation

Next, we give the experimental results of the MKC algorithm given by (45)–(47) with orientation and scale estimation. The target region (bottom row of Figs. 5–9) is marked by an ellipse with a (green) line across it representing the orientation. We compare the MKC algorithm with

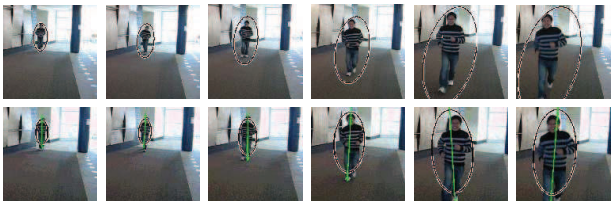

Figure 5: Human Walking Sequence-1. Frames: 1, 10, 35, 60, 72, 75.

Figure 6: Human Walking Sequence-2. Frames: 1, 50, 60, 74, 150, 170.

the method of [11] (top row of Figs. 5–9). The latter applies the mean shift algorithm to a 4D kernel, namely

$$K(x, y, \sigma, \theta) = K(x, y)K(\sigma)K(\theta) \tag{53}$$

where $K(x, y)$ is the spatial kernel, $K(\sigma)$ is the scale kernel and $K(\theta)$ is the orientation kernel. The thresholds (for both algorithms) are chosen as $\epsilon_{\mathbf{y}} = 0.7$, $\epsilon_{\phi} = 0.01$ rad. and $\epsilon_h = 0.01$. For the MKC algorithm, the average number of iterations is about 3 and the average number of Armijo Rule iterations is about 2, while for the algorithm of [11], the average number of iterations is about 8.

In Fig. 5 the person in the sequence walked quickly towards the camera, which resulted in fast scale changes. Both methods handled the scale changes very well. In Fig. 6 the target underwent large changes in both scale and orientation. The MKC algorithm tracked the target well while the algorithm of [11] failed to estimate the orientational changes and "took" non-object regions into the kernel window. In Fig. 7 the scenario is even more challenging with strong background interference. The MKC algorithm kept the target in track throughout the sequence. The tracking performance of the algorithm of [11] degraded drastically after the background interference arose and its scale estimate diverged at the end of the sequence.

In the Box sequence (Fig. 8) and Pink Cup sequence (Fig. 9), the tracker was tested for fast orientational changes. In Fig. 8 the average rotational speed was about 6°/frame and the maximum rotational speed was about 14°/frame. The MKC algorithm successfully tracked these fast orientational changes. The algorithm of [11] lost the target. In Fig. 9 we added the background interference (pink, similar to the cup) and employed the PMKC (Parallel MKC introduced in Section 4.2) algorithm with the results shown in

Figure 7: Human Walking Sequence-3. Frames: 1, 30, 37, 42, 55, 112.

Figure 8: Box Sequence. Frames: 1, 20, 30, 40, 50, 60

the middle row. The average rotational speed was about 11°/frame and the maximum rotational speed was about 18°/frame. The PMKC tracker outperformed the MKC tracker in this particularly difficult scenario with fast rotation and background interference. The performance of [11] was the worst.

6 Summary and conclusions

This paper presented a new framework for target model representation based on multiple kernel centers (MKC). Compared to the traditional model computed with a single kernel center at the centroid (SKC), the MKC model is more flexible in the target representation and more robust due to the spatial information it carries. The orientation and scale estimates are exploited from the MKC model by employing steepest gradient ascent. The proposed MKC algorithm and the mean shift approach have in common an adaptive stepsize rule, which results in a fast convergence rate. The parallel MKC algorithm was also introduced and shown to improve the tracking performance drastically.

Appendix

Proposition. If the kernel K has a convex and a monotonically decreasing profile, the sequences $\{\mathbf{y}_j\}$ given by (38) converges to $\mathbf{y}^*$, where $\rho(\mathbf{y}^*)$ is the local maximum of the similarity function defined by the Bhattacharyya Coefficient.

Proof: From (29) and (32), we have

$$\rho(\mathbf{y}^{j+1}) - \rho(\mathbf{y}^j) = \frac{1}{2}C(h_0)\sum_{i=1}^{N}\sum_{l=1}^{L} w_{i,l} \cdot \left[k\left(\left\|\frac{\mathbf{r}_l(\mathbf{y}^{j+1}) - \mathbf{x}_i}{h_0}\right\|^2\right) - k\left(\left\|\frac{\mathbf{r}_l(\mathbf{y}^{j}) - \mathbf{x}_i}{h_0}\right\|^2\right)\right]$$

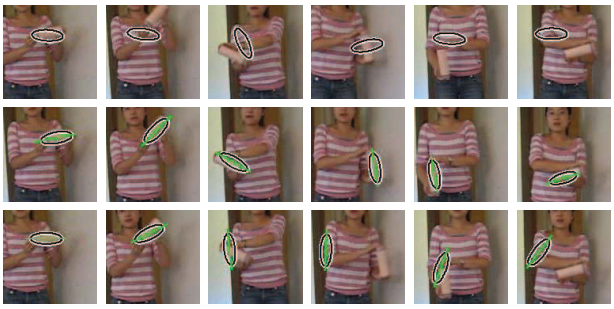

Figure 9: Pink Cup Sequence. Frames: 1, 8, 35, 80, 125, 170.

(54)

Since the kernel profile $k(x)$ is convex, the inequality

$$k(x_2) - k(x_1) \quad \geq \quad g(x_1)(x_1 - x_2) \tag{55}$$

holds, where $g(x) = -k'(x)$. Therefore, (54) becomes,

$$\begin{aligned}\rho(\mathbf{y}^{j+1}) - \rho(\mathbf{y}^j) \quad &\geq \quad \frac{C(h_0)}{h_0^2}(\mathbf{y}^{j+1} - \mathbf{y}^j)^T \\ &\cdot \sum_{i=1}^{N}\sum_{l=1}^{L}[\mathbf{x}_i - h_0\Delta\mathbf{r}_l(\phi_0)]w_{i,l}g_{i,l}{}^j \\ &+\frac{C(h_0)}{2h_0^2}(\|\mathbf{y}^j\|^2 - \|\mathbf{y}^{j+1}\|^2)\sum_{i=1}^{N}\sum_{l=1}^{L} w_{i,l}g_{i,l}{}^j \end{aligned} \tag{56}$$

by recalling (30). Using the iterations given by (38), we obtain

$$\begin{aligned}&\rho(\mathbf{y}^{j+1}) - \rho(\mathbf{y}^j) \quad \geq \\ &\frac{C(h_0)}{2h_0^2}\|\mathbf{y}^{j+1} - \mathbf{y}^j\|^2 \sum_{i=1}^{N}\sum_{l=1}^{L} w_{i,l}g_{i,l}{}^j \end{aligned} \tag{57}$$

Since profile $k(x)$ is monotonically decreasing for all $x \geq 0$ and the weight $w_{i,l}$ is nonnegative, the right term of (57) is always positive as long as $\mathbf{y}^{j+1} \neq \mathbf{y}^j$ (at least one nonzero $w_{i,l}$ by assumption). Therefore, $\rho(\mathbf{y}^j)$ is monotonically increasing for $\mathbf{y}^{j+1} \neq \mathbf{y}^j$. Since, $\rho(\mathbf{y})$ is bounded by 1, the sequence $\{\rho(\mathbf{y}^j)\}$ converges to its local maxima $\rho(\mathbf{y}^*)$ for $\mathbf{y}^{j+1} = \mathbf{y}^j = \mathbf{y}^*$. **Q.E.D.**

References

[1] G. Bradski, "Computer vision face tracking for use in a perceptual user interface," *In IEEE Workshop on Applications of Computer Vision*, 1998.

[2] D. P.Bertsekas, *Nonlinear programming, second ed*, Athena Scientific, 2003.

[3] S. Birchfield and S. Rangarajan, "Spatiograms versus histograms for region-based tracking," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 1158–1163.

[4] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysi," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 24, No. 5, pp. 603–619, 2002.

[5] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 25, No. 5, pp. 564–577, 2003.

[6] R. T. Collins, "Mean-shift blob tracking through scale space," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 234–240.

[7] Z. Fan, Y. Wu and M. Yang, "Multiple collaborative kernel tracking," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 502–509.

[8] G. D. Hager, M. Dewan and C. V. Stewart, "Multiple kernel tracking with ssd," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 790–797.

[9] A. Jepson, D. Fleet and D. Elmaraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 25, No. 10, pp. 1296–1311, 2003.

[10] V. Parameswaran and V. Ramesh, and I. Zoghlami, "Tunable kernels for tracking," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2179–2186.

[11] A. Yilmaz, "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6.

[12] A. Yilmaz, O. Javed and M. Shah, "Object tracking: A survey," *ACM Journal of Computing Surveys*, Vol 38, No. 4, pp. 1–45, 2006.